

# スキーマ設計履歴を用いた設計オーサリングツール

家富 誠敏<sup>†</sup> 有澤 博<sup>‡</sup>

<sup>†</sup>城西国際大学経営情報学部 〒283-8555 千葉県東金市求名1番地  
<sup>‡</sup>横浜国立大学大学院環境情報研究院 〒240-8501 横浜市保土ヶ谷区常盤台79-7

E-mail: <sup>†</sup>ietomi@jiu.ac.jp, <sup>‡</sup>arisawa@ynu.ac.jp

**あらまし** データベース設計において、設計者は最終結果としてデータベーススキーマを構築するが、その設計過程においては、試行錯誤の中、多くの「欠陥を含んだスキーマ」が途中段階に考案されることが多い。これら「欠陥を含んだスキーマ」と「インスタンスによる欠陥事例」を設計履歴とともに蓄積しておくことで、設計者の設計思想を後任の設計者に伝え、スキーマを変更または新たに構築する際に参考とすることができ、同種の間違いに対しては履歴との照合を行なうことにより警告を発することも可能になる。本論文では、特に履歴との照合手法について考察し、欠陥スキーマと類似するスキーマを自動的に検出し、視覚的に提示するシステムを構築する。

**キーワード** ユーザインタフェース、データの可視化、データベース設計

## Database Design Authoring Tool by using of Schema Design History

Masatoshi IETOMI<sup>†</sup> Hiroshi ARISAWA<sup>‡</sup>

<sup>†</sup> Faculty of Management and Information Sciences, Josai International University

1 Gumyo, Togane City, Chiba, 293-8555 Japan

<sup>‡</sup> Graduate School of Environment and Information Sciences, Yokohama National University

Tokiwadai 79-7, Hodogaya-ku, Yokohama, 240-8501 Japan

E-mail: <sup>†</sup>ietomi@jiu.ac.jp, <sup>‡</sup>arisawa@ynu.ac.jp

**Abstract** Although a database designer builds a database schema as a result, in the design process, many "schemas including a defect" is devised by trial and error. If "schema including a defect" and "defective example by the instance" are accumulated with the design history, a new designer can understand the design intension of a former designer, and it can be made reference, when changing a schema, or when newly constructing, moreover by matching with a history, it can also discover and warn of a same kind defect. In this paper, especially the method of matching with a defective schema in design history is considered, a schema similar to a defective schema is detected automatically, and the visual interface is presented.

**Keyword** User Interface , Information Visualization , Database Design

### 1. はじめに

データベース設計者は、実世界をモデル化した結果としてデータベーススキーマを設計するが、最終的に設計されたスキーマからは設計者がどのような考えで実世界を捉えたか、どのような意図に従ってモデル化したかという情報を推測することは困難である。

我々は、これら設計者の持つ思想や意図を、設計時の履歴（作成途中スキーマの変遷）と「途中段階のスキーマのどこに問題があったのか」を実例（インスタンス）として表現し蓄積する設計ツールを開発してきた[1]。本ツールでは、設計時に全スキーマ作成履歴を保存した上で、重要と考えられる作成途中のスキーマ

を抜き出し、各々どのような問題点があったのかを、設計者が実際にスキーマの検定に用いたインスタンスおよびそれに対する評価（どこに問題があったか等のコメント）をセットにして蓄積する。これにより、設計者以外でも設計履歴を閲覧することで、設計の各段階において設計者が考えた問題点を把握し、設計に対する考え方を理解することができる。例えば設計者の設計思想を後任の設計者に伝え、スキーマを変更または新たに構築する際に参考とすることが可能になる。

本稿では、設計履歴を単に閲覧するだけでなく、さらに再利用する方法として、現在設計中のスキーマと設計履歴とを照合することで、過去設計されたスキーマの中で最も類似しているスキーマを検索し、提示す

る手法を提案する。これにより現在設計中のスキーマと類似したスキーマが設計履歴中にあった場合、「そのスキーマのどこに問題点があったのか」「それに対してどのように改良していったのか」ということを参考にし、現在のスキーマ設計に役立てることができる。

まず、これまでの研究成果である設計履歴を蓄積する設計ツールについて概要を述べ、その後、新たな提案事項である履歴からの類似スキーマ検索手法について述べる。

## 2. 設計履歴を蓄積する設計ツール

これまで我々は設計時に設計履歴を蓄積するための設計ツールを構築してきた。

設計ツールとしては、スキーマを図的にデザインするツール[2][3][4]が数多く提案されているが、本ツールは作成操作の履歴とともに、何故そのような履歴を辿ったのかがわかるように、設計者の設計途中スキーマに対する見解（評価や問題点）を記録しておく点が他のツールにない特徴である。

本ツールでは、設計者はER図[5]を用いて視覚的に画面上に表示されたスキーマをマウス等を用いて編集することにより、スキーマ設計を行なう。スキーマ設計時に作成途中のスキーマを要所所で記録してもらうことにより、設計者が大局的にどのような試行を経て、最終的なスキーマにたどり着いたのかを保存する。

また、このとき設計者は設計途中のスキーマを検定するために、そのスキーマに対する検定用インスタンスを入力することができる(図1)。検定用インスタンスの入力は画面上のスキーマに重ねてインスタンス図を描くことによって行ない、この視覚的に表現されたインスタンス図を見ながら、そのインスタンスを用いたスキーマ検定結果を設計者のコメントやインスタンスの各部に対するカラーリング等を組み合わせることで表現し、これを蓄積することが可能である。

例えば、カラーリングによって

- ・ インスタンスの一部に青く着色  
現在のスキーマでは表現できないが、実際には表現したい内容であることを示す

- ・ インスタンスの一部に赤く着色  
現在のスキーマでは表現できてしまうが、実際には表現したくない内容であることを示す

というようにインスタンスの着目すべき部位を視覚的に表し、そこにコメント(どうして問題があるのか等)を記述しておくことで、設計者が設計時にどのような考えをスキーマに抱いていたのかを蓄積する。

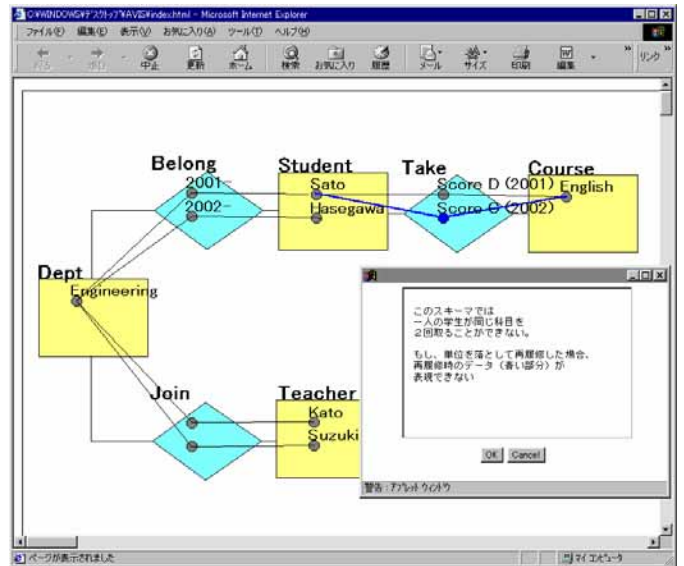


図1 検定インスタンス

設計時にこれら設計履歴を蓄積しておくことによって、設計者以外の利用者が後に以下のような操作を行なうことができる。

- ・ 設計時における設計途中スキーマの推移を有向グラフによる表現や実際の設計操作を簡易アニメーションで表現したもによって捉えることができる
- ・ スキーマの推移を見ながら、必要に応じて各途中スキーマに対し、どのような検定が行われたのかを視覚化された検定インスタンスおよびそこに付加された検定結果から知ることができる。

これにより、設計者が設計時にどのような問題に直面し、その問題に対してどのような解決を図っていったかを把握することが可能になる。

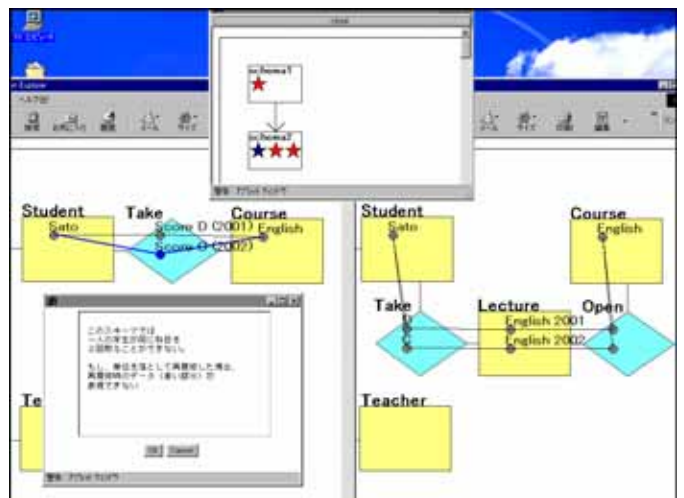


図2 設計プロセスの表示

図2は実際にデータベースを設計した際の設計履歴を画面に表示した例である。画面上中央にあるのが、スキーマ履歴を表したグラフであり、ここでは設計者が schema1 を参考にして schema2 を作成していった場合の例であることを表している。このスキーマグラフ上で、スキーマを表す矩形や検定インスタンスを表す星型をマウスでクリックすると、そのスキーマやインスタンスが画面に図で表示される。この例では、2つの検定インスタンスを画面に表示させているが、画面左が schema1、画面右が schema2 の検定インスタンスである。ここでは schema1 に付加された検定インスタンスおよびコメントによって、設計者がどのような問題点に着目していたのかが示され、schema2 に付加された検定インスタンスによって、設計者がこの問題をどのように解決したのかが実例で示されている。さらに詳細を知りたい場合には、設計者の行なったスキーマ変更操作を細かくアニメーションで見えていくこともできる。

### 3. 類似スキーマの検索

設計履歴を蓄積することは、設計者の持つ思想・意図を他人に伝えることが目的であるが、それが役立つケースの一つとして、設計へのフィードバックが考えられる。すなわち、新たにデータベーススキーマを設計する際（または既に設計されたスキーマを変更する際）他の設計者がどのような思想・意図で設計を行なったかを把握し、設計に役立てるという利用法である。しかし、履歴を閲覧することで他の設計者の意図を理解することは、履歴に含まれるスキーマが膨大である場合や、数多くの設計者による履歴が蓄積されているような場合には非常に困難であると考えられる。特に典型的なデータベースであれば、過去において様々な設計者が数多くの設計を行なっている場合も想定され、単に履歴を閲覧するだけの機能では、設計時に設計履歴を利用するために十分ではない。

ここでは、多くの設計者による膨大な数の設計履歴の中から、現在の設計に最も必要な情報を効率よく提供する手段として、類似スキーマを検索し提示する手法を提案する。これは設計者が現在設計しているスキーマと最も類似しているスキーマを設計履歴の中から検索し提示することで「そのスキーマにどのような問題点があったのか」「それをどのように解決していったのか」等の情報を効率良く提供し、設計の参考にしてもらうことを目的としている。

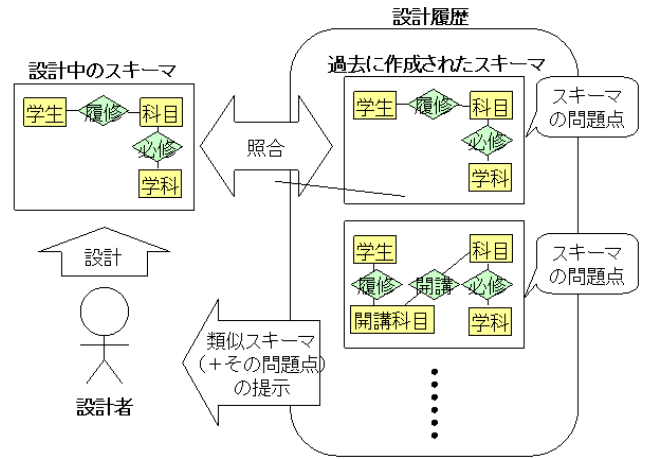


図3 類似スキーマ提示の概念

#### 3.1. ER図におけるスキーマの類似

スキーマを ER 図で表現した際、ここでは次の条件を全て満たすスキーマを設計の参考となる類似スキーマとして検索することを考える。

ER 図の全体または一部において

1. トポロジーが一致している
2. 対応する entity-type の性質が一致している
3. 対応する relationship-type の性質が一致している

図4は類似スキーマの一例である。スキーマ1とスキーマ2は entity-type および relationship-type のつながり方が同様であり、entity-type の「学科」と「DEPT」、「学生」と「STUDENT」、「科目」と「SUBJ」は名称こそ違っても同じ性質（同様の概念を表すもの）であり、また relationship-type 「所属」と「BELONG」、「履修」と「TAKE」も同じ性質（同じ概念の entity-type を同じ対応関係で結び付けている）である。ゆえにスキーマ1とスキーマ2は条件1～3を満たし、類似している。

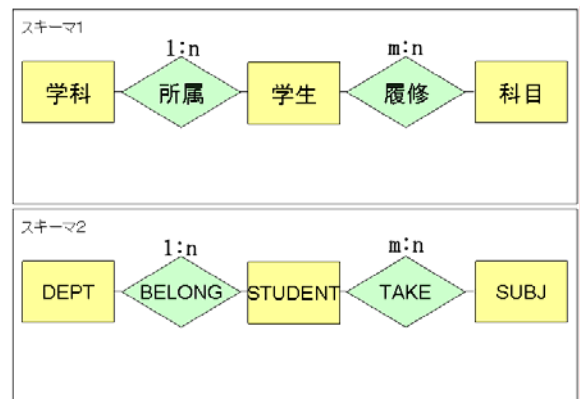


図4 類似スキーマの例1

また図5も類似スキーマの例である。スキーマ1の entity-type 「学生」「科目」とこれを結ぶ「履修」が、スキーマ2の「STUDENT」「SUBJ」と「TAKE」が条

件 1 ~ 3 で一致しており、スキーマ 1 の一部とスキーマ 2 が類似している。設計の参考となる場合は、このように部分的に類似しているスキーマが大半であると考えられるので、部分的な類似も考慮する必要がある。

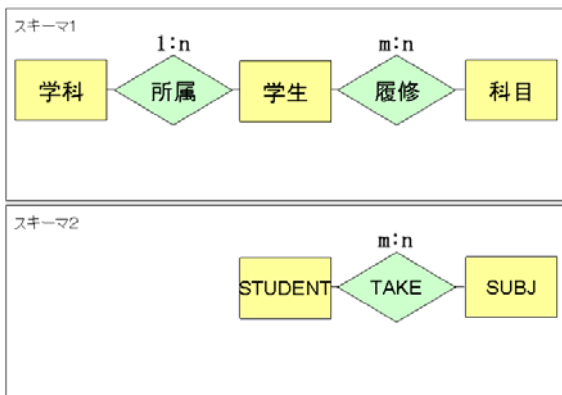


図 5 類似スキーマの例 2

### 3.2. アイコン画像を用いた entity-type の類似度判定

類似スキーマを検索するとき、最も大きな問題となるのが entity-type の一致判定である。entity-type は名称が異なっても、その性質（表す概念）が一致している場合が存在するため、単に名称を用いて一致判定を行なうことができない。性質が同一であることを判定する手法としては、予め entity-type に様々なメタ情報を付加しておき、それらメタ情報を用いて類似判定することが考えられる。その際、最も簡単な方法は、設計時に entity-type を定義する際、設計者の手によりメタ情報を入力してもらうことであるが、キーワード的な情報を設計者に付加させることは、設計者毎に表現の違いがあり、設計者の異なる設計履歴に対して類似判定を行なう上ではあまり好ましくない。

そこで、設計者の表現の違いをできる限り少なくするため、次のような方法を用いた。

1. 設計者にはスキーマ設計時に検定インスタンスを入力してもらうが、この際、各 entity アイコンに可能な限り画像を貼り付けてもらう
2. アイコンに貼り付ける画像は予め用意されたイメージクリップの中から最もその entity を表すのに適当と思われるものを選択してもらう
3. 個々のイメージにはイメージを象徴するキーワードが複数個付加されており、entity-type 内の全 entity に付加されたイメージのキーワードを集計し entity-type のメタ情報として格納しておく
4. entity-type の持つイメージキーワードの集計（キーワード毎の出現回数）が近似したヒストグラム形状を持つ entity-type 同士を類似と判定する

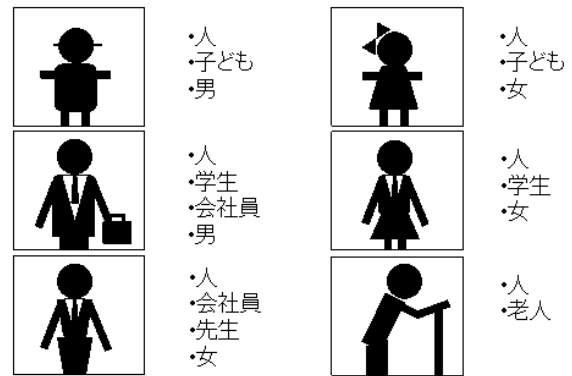


図 6 イメージクリップと付加キーワードの一例

入力された検定インスタンスのアイコン画像を手がかりに entity-type の類似判定を行なうことで、たとえ名称が異なっても同様の性質を持つ entity-type を判別する。

### 3.3. 試作システムによる類似スキーマの提示

図 7 は試作システムの実行画面である。左上が設計中のスキーマであり、右下が設計履歴から検索された類似スキーマである。各 entity-type や relationship-type の名称が異なっても、インスタンスに付加された画像同士の類似度を画像が持つキーワード群の照合（頻度ヒストグラムの照合）により判定し、類似スキーマを検索する。この例では「学生」および「STUDENT」は“人”や“学生”というキーワードを持つ画像を含み、「科目」と「COURSE」は“本”や“教科書”というキーワードを持つ画像をとも含むため、一致する entity-type として判定され、類似スキーマとして提示される。これら判定のためには、イメージクリップの画像とそのキーワードの選定が重要であり、適切な画像およびキーワードについて現在分析中である。

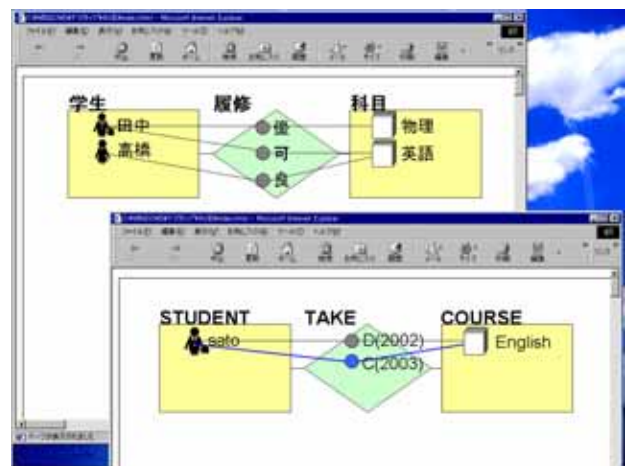


図 7 試作システムによる類似スキーマの表示

#### 4. まとめ

本研究では、蓄積された設計履歴の中から、現在設計中のスキーマに最も類似したスキーマに関する情報を提示することにより、新たな設計に役立てることを目的とし、類似スキーマを検索する手法を提案した。

現在、まだ試作システムの開発中であり、システム開発および実際に利用した上での評価などをこれから行っていく予定である。

#### 文 献

- [1] 家富 誠敏, 有澤 博, :“ 設計意図を表現可能なデータベース設計手法とその設計支援ツール ”, 第 14 回データ工学ワークショップ(DEWS2003)論文, 2003
- [2] A.Grau, et al :“ The TROLL Approach to Conceptual Modelling : Syntax, Semantics and Tools ”, Proceedings of the International Conference on Conceptual Modelling, pp.277-290, 1998.
- [3] Sudha Ram, et al :“ DISTIL: A Design Support Environment for Conceptual Modeling of Spatio-temporal Requirements ”, Conceptual Modeling - ER2001, Springer LNCS 2224, pp.70-83, 2001.
- [4] Virginie Detienne, Jean-Luc Hainaut : “ CASE Tool Support for Temporal Database Design ”, Conceptual Modeling - ER2001, Springer LNCS 2224, pp.208-224, 2001.
- [5] P. P. Chen : “ The Entity-Relationship Model : Toward a Unified View of Data ”, ACM Transactions on Database Systems, Vol.1, No.1, 1976.